Spectral Clustering with Links and Attributes

Jennifer Neville
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
jneville@cs.umass.edu

Micah Adler
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
micah@cs.umass.edu

David Jensen
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
jensen@cs.umass.edu

ABSTRACT

If relational data contain *communities*—groups of inter-related items with similar attribute values—a clustering technique that considers attribute information and the structure of relations simultaneously should produce more meaningful clusters than those produced by considering attributes alone. We investigate this hypothesis in the context of a spectral graph partitioning technique, considering a number of hybrid similarity metrics that combine both sources of information. Through simulation, we find that two of the hybrid metrics achieve superior performance over a wide range of data characteristics. We analyze the spectral decomposition algorithm from a statistical perspective and show that the successful hybrid metrics exaggerate the separation between cluster similarity values, at the expense of increased variance. We cluster several relational datasets using the best hybrid metric and show that the resulting clusters exhibit significant community structure, and that they significantly improve performance in a related classification task.

Categories and Subject Descriptors

I.5.3 [**Clustering**]: Pattern Recognition

Keywords

Clustering, Relational Learning, Spectral Analysis

1. INTRODUCTION

Spectral clustering techniques, which partition data into disjoint clusters using the eigenstructure of a similarity matrix, have been successfully applied in a number of domains, including image segmentation [19] and document clustering [5]. Finding an optimal partition is in general NP complete, but the eigenvectors of the matrix provide some information that can be used to guide an approximate solution. Experimental evidence has shown this heuristic approach often works well in practice and has prompted further investigation into the properties of spectral clustering. Recent findings—facilitated by a long history of work in spectral graph theory (e.g., [2])—include a connection to random walks [13] and preliminary performance analysis [10, 16]. In this paper, we investigate methods of adapting spectral clustering techniques to relational domains.

The goal of this work is to find *communities* in relational data represented as an attributed graph $G = (V, E, X)$, where the nodes V represent objects in the data (e.g., genes), the edges E represent relations among the objects (e.g., interactions), and the attributes X record data about each object (e.g., localization). Community clusters identify groups of objects that have similar attributes and are also highly inter-related. For example in genomic data, a group of genes with similar attributes and many common interactions may all be involved in a similar function in the cell. The underlying assumption is that there is a latent cluster variable that influences both the attribute values intrinsic to objects and the relationships among objects. In particular, objects are more likely to link to other objects in the same cluster than objects in other clusters, and pairs of objects within a cluster are more likely to have similar attribute values than pairs spanning different clusters. A clustering algorithm that examines both link structure and attributes simultaneously should be more robust to noise than methods examining attribute or link information in isolation.

There has been little work applying spectral techniques to relational domains with a combination of link and attribute information. Existing techniques use either: (1) a *complete* graph where attribute similarity is calculated for all $n \times n$ pairs of objects (e.g., [16]), or (2) a nearest neighbor graph, where attribute similarity is calculated for $n \times d$ pairs of objects—each object is connected to a fixed number (d) of other objects determined by spatial locality (e.g., [19]). Our work differs in that we are trying to incorporate the heterogeneous relational structure into the similarity metric.

The similarity metric, used to populate the similarity matrix, provides a means to extend spectral techniques to new domains. However, the success of spectral clustering techniques depends heavily on the choice of metric. There has been some research into *learning* the correct similarity function from labeled data (e.g., [1]), but for domains where the correct clustering is unknown, design has been approached in a relatively ad-hoc manner. This leaves us with little guidance as to how to incorporate link and attribute information into a metric for relational domains. This work investigates the design of similarity metrics that incorporate multiple

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Spectral Clustering with Links and Attributes	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Massachusetts,Department of Computer Science,Amherst,MA,01003-9264**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

If relational data contain communities?groups of inter-related items with similar attribute values-a clustering technique that considers attribute information and the structure of relations simultaneously should produce more meaningful clusters than those produced by considering attributes alone. We investigate this hypothesis in the context of a spectral graph partitioning technique, considering a number of hybrid similarity metrics that combine both sources of information. Through simulation, we find that two of the hybrid metrics achieve superior performance over a wide range of data characteristics. We analyze the spectral decomposition algorithm from a statistical perspective and show that the successful hybrid metrics exaggerate the separation between cluster similarity values, at the expense of increased variance. We cluster several relational datasets using the best hybrid metric and show that the resulting clusters exhibit significant community structure, and that they significantly improve performance in a related classification task.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

sources of information and identifies the characteristics that underlie successful metrics.

Specifically, we analyze the *normalized cut* (NCut) spectral partitioning algorithm [19] from a statistical perspective. For the special case of bi-partitioning, we show that as cluster size $\to \infty$, the spectral decomposition will include an eigenvector that is piecewise constant, with respect to the clusters, for any similarity metric where the average intra-cluster similarity differs from the average inter-cluster similarity. If the eigenvector associated with the 2^{nd} smallest eigenvalue of the similarity matrix is piecewise constant, the spectral partitioning will be exact [19]. Next, we empirically evaluate the effect of finite cluster sizes using synthetic data. We show that: (1) decreasing variance of cluster similarities, and increasing separation of similarities, both improve the ordering of the eigenvector with respect to the clusters, and (2) increasing the separation of cluster similarities has a greater impact on algorithm performance when the NCut objective function is used. This indicates that a metric that increases variance in order to better separate the cluster similarities will perform better over a wider range of conditions. Based on these results, we propose a hybrid similarity metric for relational data that incorporates link and attribute information, and we evaluate performance on several relational datasets. We show that resulting clusters exhibit significant *community* structure and demonstrate significant performance gains when using the resulting clusters in a related classification task.

2. SPECTRAL CLUSTERING

Spectral clustering originated with graph partitioning techniques that exploit the connection between eigenvectors and algebraic properties of a graph (e.g., [6, 7]). Recently, Shi and Malik [19] presented a new clustering algorithm that uses spectral partitioning to optimize the NCut objective function. We investigate the application of this algorithm to relational domains through the use of similarity metrics that incorporate link and attribute information.

The NCut algorithm of [19] clusters datasets through eigenvalue decomposition of a similarity matrix. The algorithm is a divisive, hierarchical clustering algorithm, which takes a graph $G = (V, E)$, a set of k attributes $\mathbf{X} = \{\mathbf{X^1}, \cdots, \mathbf{X^k}\}$, where $\mathbf{X^k} = \{x_i^k : v_i \in V\}$, and a similarity function S, where $S(i, j)$ defines the similarity between $v_i, v_j \in V$, and recursively partitions the graph as follows:

Let $\mathbf{W}_{N\times N} = [S(i, j)]$ be the similarity matrix and let $\mathbf{D}$ be an $N \times N$ diagonal matrix with $d_i = \sum_{j\in V} S(i, j)$. Solve the eigensystem $(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$ for the eigenvector $\mathbf{x}_1$ associated with the 2^{nd} smallest eigenvalue λ_1. Consider m uniform values between the minimum and maximum value in $\mathbf{x}_1$. For each value m: bipartition the nodes into (A, B) such that $A \cap B = \emptyset, A \cup B = V$, and $\forall v_a \in A\, x_{1a} < m$, and calculate the NCut value for the partition, $NCut(A, B) = \frac{\sum_{i\in A, j\in B} S(i,j)}{\sum_{i\in A} d_i} + \frac{\sum_{i\in A, j\in B} S(i,j)}{\sum_{j\in B} d_j}$. Partition the graph into the (A, B) with minimum NCut. If $stability(A, B) \le c$, recursively repartition A and B.[1]

[1]We use the stability threshold proposed in [19] where the stability value is the ratio of the minimum and maximum bin sizes, after the values of $\mathbf{x}_1$ are binned by value into m bins. All the experiments in this paper used the settings: $m = \lceil log_2(N)+1 \rceil$, and $c = 0.06$. Sensitivity analysis on synthetic data shows $c = 0.06$ to be a conservative threshold, returning clusters with high precision but low recall.

It takes $O(n^3)$ operations to solve for all eigenvalues of an arbitrary eigensystem. However, $O(|E|)$ approximate algorithms exist [10], and if the weight matrix is sparse, $O(n^{1.4})$ Lanczos algorithms can be used to compute the solution [18]—for this reason, similarity metrics that produce sparse matrices are preferable.

Our *hybrid* metrics calculate the similarity between objects i and j through a weighted combination of attribute and link information: $S(i, j) = \alpha \cdot \frac{1}{k} \sum_k s_k(i, j) + (1 - \alpha) \cdot l$, where $s_k(i, j) = 1$ if $x_i^k = x_j^k$ and 0 otherwise, and $l = 1$ if $e_{ij} \in E$ or $e_{ji} \in E$, and 0 otherwise.

When $\alpha = 1$, we refer to the metric as *AttrOnly*. When $\alpha = 0$, we refer to the metric as *LinkOnly*. These metrics are included as baselines—one for data clustering techniques that ignore link information, and the other for graph partitioning techniques that ignore attribute information. *AttrOnly* calculates similarity by counting the number of attribute values objects i and j have in common (scaled by k so the maximum similarity is 1). *LinkOnly* uses the relational structure as a measure of similarity.

When $\alpha = \frac{k}{k+1}$, we refer to the metric as *LinkAsAttr*. This approach is an obvious way to include relational information—links are incorporated as a match on the $(k+1)^{th}$ attribute. With no prior domain knowledge, we have no reason to expect that link structure contains more information than attribute values. However, link structure is often central in relational domains—for example, in a graph of hyperlinked web documents, we expect a link to confer more information about topic clustering than a match on a single word for two pages. To better exploit the relational information, we set $\alpha = \frac{1}{2}$. This metric, referred to as *WtLinkAttr1*, combines the link and attribute information uniformly—high similarity indicates that two objects are related *or* have a number of attribute values in common.

In sparse relational graphs, the expected intra-cluster link similarity will be less than one, even if the links are perfectly correlated with cluster membership. In this case, if the link and attribute information are combined uniformly (e.g., *WtLinkAttr1*), or if the attributes are given proportionally more weight (e.g., *LinkAsAttr*), noise in the attributes can drown out a strong link signal. An approach that gives the link information proportionally more weight (e.g., $\alpha > \frac{1}{2}$) may achieve better performance. In practice we will not know how to scale the link information to combine the two sources of information equally. However, for the synthetic experiments discussed in the next section, we know the maximum edge probability is 0.2 so setting $\alpha = \frac{1}{6}$ equalizes the attribute and link signals. When $\alpha = \frac{1}{6}$, we refer to the metric as *WtLinkAttr2*. Although we will not know the scaling factor in practice, we include this metric to test the conjecture that the poor performance of *WtLinkAttr1* is due to the relatively weak link signal being combined uniformly with the attribute signal.

When $\alpha = l$, we refer to the metric as *LinkAsFilter*. It cal-

culates similarity by weighting the existing edges of G with the *AttrOnly* metric. Objects that are not directly related have a similarity of 0 regardless of their attribute values. A high similarity score indicates that two objects are related *and* have a number of attribute values in common. This approach incorporates both sources of information while maintaining the sparsity of the relational data graph so the algorithm can use efficient eigensolver techniques.

3. ALGORITHM ANALYSIS

The recursive nature of the algorithm complicates analysis of higher-order partitioning, so we restrict our attention to the (simpler) case of a single bipartitioning of the graph. Finding an optimal partition, which minimizes the NCut criterion, is an NP-hard problem [19]. However, [19] shows that when there is a partition (A, B) of V such that the 2^{nd} smallest eigenvector $\mathbf{x}_1$, of the eigensystem $(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$, is *piecewise constant* with respect to a partition (A, B): $\mathbf{x}_{1i} = \alpha, i \in A$, and $\mathbf{x}_{1i} = \beta, i \in B, \beta \neq \alpha$, then (A, B) is the optimal partition—it minimizes the NCut criterion and $\lambda_1 = NCut$.

Recent analysis has focused on achieving a more thorough understanding of the conditions under which $\mathbf{x}_1$ will be piecewise constant. Meila and Shi [13] outline a set of conditions under which the spectral algorithm will return an exact partitioning, showing that the spectral problem formulated for NCut is equivalent to the eigenvectors/values of the stochastic matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. The authors connect spectral clustering to Markov random walks, showing that $\mathbf{P}$ will have an eigenvector that is piecewise constant w.r.t. a partition (A_1, A_2) iff $\mathbf{P}$ is *block-stochastic* w.r.t. (A_1, A_2). Here, block-stochastic means that the underlying Markov random walk can be viewed as a Markov chain with state space $\Delta = (A_1, A_2)$ and transition probability matrix $\mathbf{R} = [\mathbf{P}_{ss'}]_{s,s'=1,2}$, where for $s, s' = 1, 2$, $\sum_{j \in A_{s'}} \mathbf{P}_{ij}$ is constant $\forall i \in A_s$, and $\mathbf{P}_{ss'} = \sum_{j \in A_{s'}} \mathbf{P}_{ij}$ for any $i \in A_s$. This shows that spectral clustering groups nodes based on the similarity of their transition probabilities to subsets of the graph.

There has been little analysis of the impact of non-constant transition probabilities on algorithm performance. Empirical evidence indicates that the algorithm finds good partitions even when the transition probabilities are far from constant. Ideally, we would like to characterize the conditions necessary for optimal performance and bound algorithm performance otherwise. As a first step, we analyze asymptotic performance for non-constant intra- and inter-cluster transition probabilities.

If we assume a generative model of the data where a latent cluster variable (A_1, A_2), determines the attribute values intrinsic to the objects and the relationships among objects, we can analyze the similarity metric $S(i, j)$, and each entry in $\mathbf{W}$, as a random variable. Consider the entries of row i. The entries $\mathbf{W}_{ij}, \mathbf{W}_{ik}$ are not independent because the similarity values are both based on node i. However, conditioned on the state of i (e.g., attribute values of i), the entries are independent random variables since the state of j is independent of the state of k. As a result, the entries of row i can be viewed as independent random variables. With this model we can show that any similarity metric will produce piecewise constant eigenvectors in the limit.

Theorem: *Let $\Delta = (A_1, A_2)$ be a partition of V. Let the function $S(i, j)$ define the similarity measure between $v_i, v_j \in V$. If, $\forall i, j, k$, $S(i, j)$ is conditionally independent of $S(i, k)$ given node i, and $E[\mathbf{P}_{11}]E[\mathbf{P}_{22}] \neq E[\mathbf{P}_{12}]E[\mathbf{P}_{21}]$ then, $\mathbf{P}$ has an eigenvector that will converge to piecewise constant w.r.t. Δ as $|A_1|, |A_2| \to \infty$.*

We provide the intuition for the proof here and refer the reader to Appendix A for details. If we view the entries of $\mathbf{W}$ as random variables, the normalized values in $\mathbf{P}$ are also random variables (i.e., the entries in $\mathbf{W}$ divided by a row sum of random variables). The total intra- and inter-cluster transition probabilities in $\mathbf{P}$ (e.g., $\sum_{j \in A_{s'}} \mathbf{P}_{ij}$) then correspond to the ratio of two sums of random variables. Since the transition probabilities are composed of sums of independent random variables, as cluster size $\to \infty$, the intra- and inter-cluster transition probabilities will converge to the same value for all nodes in each cluster. Therefore an eigenvector of the similarity matrix will converge to piecewise constant w.r.t. (A_1, A_2), provided the intra- and inter-cluster means (e.g., $E[\mathbf{P}_{11}]$, $E[\mathbf{P}_{12}]$) are distinguishable.

This analysis indicates that all metrics will perform equally in the limit. We expect however, that finite sample performance will vary based on the characteristics of the metrics. In particular, we expect that performance will be influenced by the mean and variance of the intra- and inter cluster transition probabilities. We demonstrate the impact of the transition probability distributions below, using synthetic data experiments.

4. SYNTHETIC DATA EXPERIMENTS

In order to identify the situations where we can expect each of the similarity metrics to perform well, we evaluate algorithm performance on synthetic data sets for which the correct clustering is known. This facilitates analysis over a wide range of conditions.

4.1 Synthetic Data

Our synthetic data sets are undirected, connected graphs $(G = (V, E))$ where nodes correspond to objects and edges correspond to relations among objects. Unless otherwise indicated, $|V| = 200$. A binary label, $C = \{+, -\}$, is used to represent cluster membership; labels are assigned randomly to each object with $P(+) = 0.5$. Each object has five binary attributes, where the attribute values are assigned randomly given the object's cluster label. Edges are added to the graph by considering each pair of objects in V independently, and adding edges randomly given the cluster labels of the two objects.

The experiments record algorithm performance while varying both attribute and link association. Within each level of correlation, all five attributes were generated with the same probability: $P_+ = P(A = 1|C = +) = \{0.50, 0.55, \ldots, 0.95, 1.0\}$, $P_- = P(A = 1|C = -) = 1.0 - P_+$. The symmetry in attribute parameters simplifies the analytical analysis but it is not necessary for algorithm correctness. Intra-cluster and inter-cluster links were generated with the following range of probabilities: $P^l_{in} = P(e_{ij}|C_i = C_j) = \{0.10, 0.12, \ldots, 0.18, 0.20\}$, $P^l_{out} = P(e_{ij}|C_i \neq C_j) = 0.2 - P^l_{in}$. Here the range of probabilities, and symmetry, was chosen to produce a graph with

approximately 10% of the $n(n-1)/2$ possible edges. This level of linkage is comparable to the levels of sparsity we have observed in real-world relational data sets.

4.2 Metric Performance

We measured the accuracy of the six metrics across the range of attribute and link probabilities described above. Figure 1 reports the accuracy of the clusterings returned by the similarity metrics, averaged over 100 trials at each setting. Note that the bottom, foremost corner of each plot represents completely random link and attribute information, where no metric should do better than 0.5.

LinkOnly and *AttrOnly* performance is as expected—they perform well when the link, or respectively attribute, signal is moderate to high, but poorly otherwise. The *LinkAsAttr* and *WtLinkAttr1* results are comparable to *AttrOnly*. However, the *LinkAsFilter* and *WtLinkAttr2* metrics achieve perfect accuracy over a wide range of conditions, with *LinkAsFilter* covering more space than *WtLinkAttr2*. These metrics should yield good results in datasets where either the links or the attributes are moderately correlated with the clusters. However, they do not always perform as well as *LinkOnly* and *AttrOnly*. Consider the *LinkOnly* results when link correlation is moderate and attribute correlation is low—both hybrid metrics achieve significantly lower accuracy than would be achieved considering links in isolation. Similar behavior is apparent for the *AttrOnly* metric, but notice that the effect is more pronounced in this situation. This indicates that the two metrics rely more heavily on link information and illustrates the tradeoff for utilizing both sources of information—the additional information increases variance, which will impair performance in some situations, in exchange for better coverage of the space.

4.3 Performance Analysis

LinkAsFilter and *WtLinkAttr2* achieve superior performance over a wide range of data characteristics, but what is the mechanism by which this occurs? Following our analysis in section 3, we hypothesize that metric performance is influenced by intra- and inter-cluster transition probabilities. We conjecture that the algorithm will be able to distinguish clusters, if the distributions of intra- and inter-cluster transition probabilities are *separable*, where separation depends on the mean and variance of the transition probabilities.

Given our data generation parameters, we can calculate intra- and inter-cluster mean transition probabilities analytically. Recall that our data generation process produces the same distribution for each cluster, and furthermore, we know that the transition probabilities in $\mathbf{P}$ are normalized to sum to one. This means we can examine $\mu_{P_{in}} = E[\mathbf{P}_{in}]$ from a single set of distributions, $\mu_{P_{in}}$ and $\mu_{P_{out}}$. When $\mu_{P_{in}} = 1.0$ there is maximal separation between the two clusters; $\mu_{P_{in}} = 0.5$ corresponds to no separation.

Figure 2 graphs $\mu_{P_{in}}$ vs. attribute/link correlations. The shapes of the graphs are quite similar to the accuracy graphs in figure 1, indicating a strong relationship between mean separation and algorithm performance. However, the areas where we observe perfect performance (i.e., accuracy = 1.0) do not necessarily correspond to maximum mean separation (i.e., $\mu_{P_{in}} \leq 1.0$). This illustrates a difference between the *LinkAsFilter* and *WtLinkAttr2* metrics—$\mu_{P_{in}}$ is significantly higher on average for the *LinkAsFilter* metric.

To examine the effect of $\mu_{P_{in}}$ on algorithm performance, we analyzed the data from all metrics concurrently. Figure 3a graphs $\mu_{P_{in}}$ vs. accuracy for the experiments reported above, combining results from all the metrics in the same graph. There is a clear relationship between $\mu_{P_{in}}$ and accuracy (corr= 0.849, $p \ll 0.05$)—accuracy is consistenly high for $\mu_{P_{in}} > 0.675$ and consistently low otherwise. We looked at the association between $\mu_{P_{in}}$ and the eigenvector values in $\mathbf{x}_1$ using a number of different measures of eigenvector stability. Only one measure showed a clear relationship to $\mu_{P_{in}}$—a measure of the quality of the ordering in the (sorted) eigenvector, which looked at the sorted eigenvector and recorded the maximum accuracy possible from the set of m possible partition values considered by the algorithm. The linear search for an optimal partition (in the NCut algorithm) should not be adversely affected by degradation of piecewise constancy unless the degradation also affects the ordering of objects' eigenvector values. If the maximum accuracy is low, this indicates disorder in the eigenvector. The *evector ordering* measure is graphed against $\mu_{P_{in}}$ in figure 3b. It shows that decreasing $\mu_{P_{in}}$ results in a disordering of the eigenvector values. These results explain the high accuracy results—for $\mu_{P_{in}} > 0.675$ there is little disorder in the eigenvector.

Figure 3c graphs evector ordering vs. accuracy. There is a strong correlation between evector ordering and accuracy, but there are also a significant number of trials with very little disorder that achieve only low accuracy. This effect is explained by figure 3d, where we graph the precision of the smallest cluster returned by the algorithm. This shows that when the eigenvector is ordered correctly but the algorithm only achieves low accuracy, it is because the algorithm prefers to separate a small, but pure, cluster from the rest of the graph. Why does the algorithm break off small, high-precision clusters even when the eigenvector ordering is correct? This is not a spurious effect due to consideration of only a small number of thresholds (e.g., m values). It remains consistent even when we set $m = N$. We discuss reasons for this effect below.

We have shown that mean separation affects algorithm performance through the ordering of the objects' eigenvector values, but how does variance interact with mean separation to degrade performance? Figures 4a-b graph the same variables as figure 3a, but for a set of experiments with $|V| = 500$, and $|V| = 50$. This illustrates the impact of decreased, and increased, variance in the transition probabilities—increasing variance impairs performance for all $\mu_{P_{in}}$, but decreasing variance only improves performance for $\mu_{P_{in}} > 0.675$. This is contrary to our expectation that decreased variance would improve performance by increasing the separation between cluster transition probabilities. However, this effect is due to the NCut optimization, not the ordering of the eigenvector values. Figure 4c shows a box plot of evector ordering as a function of sample size, for the set of trials with $\mu_{P_{in}} < 0.675$. Except for the smallest sample size, where we see higher accuracy due to chance alone, the mean ordering value is monotonically increasing with sample size. Figure 4d graphs accuracy results for the

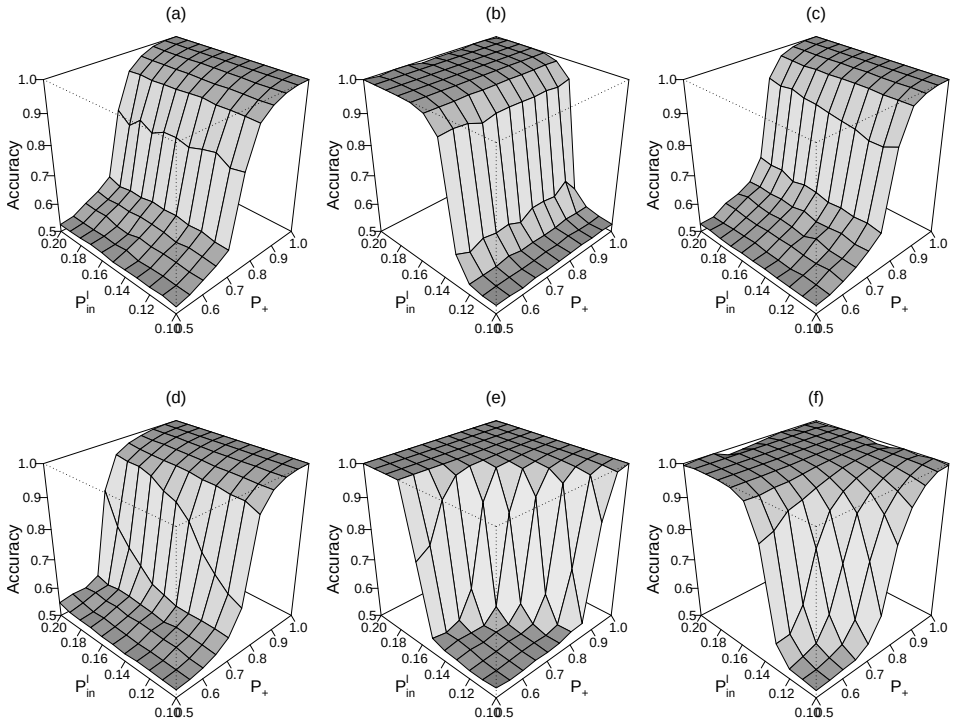

Figure 1: Cluster accuracy of metrics on synthetic data: (a) *AttrOnly*, (b) *LinkOnly*, (c) *LinkAsAttr*, (d) *WtLinkAttr1*, (e) *WtLinkAttr2*, and (f) *LinkAsFilter*.

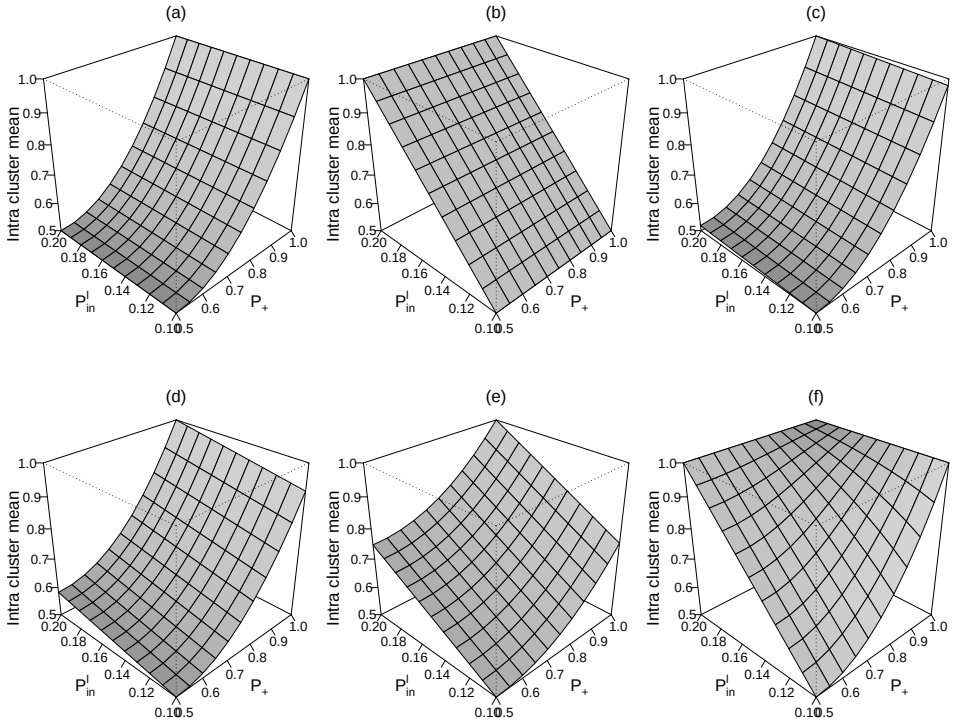

Figure 2: Intra-cluster means of metrics for synthetic data: (a) *AttrOnly*, (b) *LinkOnly*, (c) *LinkAsAttr*, (d) *WtLinkAttr1*, (e) *WtLinkAttr2*, and (f) *LinkAsFilter*.

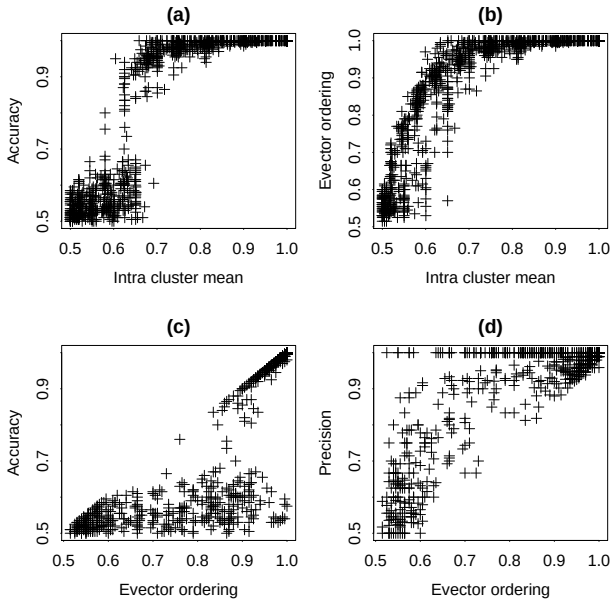

Figure 3: Analysis of intra-cluster mean on algorithm performance: (a) 200 objects, (b) $\mu_{P_{in}}$ vs. ordering, (c) ordering vs. accuracy and (d) precision.

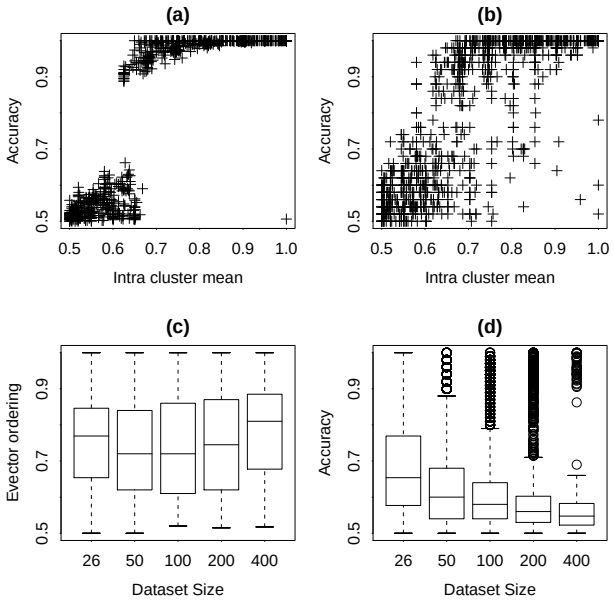

Figure 4: Analysis of intra-cluster variance on algorithm performance: (a) 500 objects, (b) 50 objects, (c) ordering and (d) accuracy for settings with $\mu_{P_{in}} < 0.675$.

same sample, showing that the algorithm converges to low accuracies as sample size increases. Maximizing the NCut criterion causes the algorithm to consistently prefer high precision over high accuracy when the separation between intra- and inter-cluster transition probabilities is low (i.e., $\mu_{P_{in}} < 0.675$). This indicates that metrics with low $\mu_{P_{in}}$ should not be combined with the NCut criterion.

It is now clear that the *WtLinkAttr2* and *LinkAsFilter* metrics achieve their good performance due to high $\mu_{P_{in}}$, but what do they tradeoff for this increased separation? Figure 5a graphs a box plot of $\mu_{P_{in}}$ for each metric individually. This is a one-dimensional summary of the data in figure 2, which again illustrates that the $\mu_{P_{in}}$ is significantly higher for the *LinkAsFilter* metric on average. Figure 5b graphs a box plot of the variance of P_{in} for each metric. This shows that *LinkAsFilter* trades off higher variance for increased mean separation. Figure 4c-d graphs the performance of *WtLinkAttr2* and *LinkAsFilter* for $|V| = 50$. Compare this to figure 1 to see that performance degradation is not uniform across metrics. The *LinkAsFilter* metric is adversly affected over a wider range of data conditions. This illustrates the primary distinction between *LinkAsFilter* and *WtLinkAttr2*. The *LinkAsFilter* metric reduces the amount of information it uses in order to increase the mean separation between the clusters. Because it is filtering the attribute information through the existing edges of the graph, it throws away both useful and noisy data and increases the variance of the transition probabilities. If the sample size is large enough to withstand this increase in variance, then the metric will produce superior clusterings. However, when the sample size is low, the filter can do more harm than good. For example, filtering through the existing edges may disconnect a previously connected cluster. In these situations, it may be best to use the *WtLinkAttr2* metric, which suffers less from increased variance and still performs well over a wide range of data characteristics. However, since we do not know how to set α for *WtLinkAttr2* in practice, and because *LinkAsFilter* offers the opportunity to use efficient eigensolver techniques, we focus on *LinkAsFilter* for our empirical data experiments.

5. EMPIRICAL DATA EXPERIMENTS

The experiments reported below are intended to evaluate two assertions. The first claim is that the *LinkAsFilter* clustering approach can be used to find groups of items with similar attribute values and high inter-connectedness. We evaluate this claim by comparing the clusters produced by the *LinkAsFilter* metric to randomly generated clusters of the same size, evaluating intra-cluster attribute similarity and intra-cluster linkage.

The second claim is that the *LinkAsFilter* clustering approach finds meaningful clusters. Evaluating clusterings of datasets for which there is no right answer is a difficult task. One approach is to present the resulting clusters for user examination. For this type of subjective evaluation, we include example cluster members from two real-world datasets. Another, more objective, approach is to examine cluster utility by evaluating the cluster labels ability to improve a related classification task. We evaluate three approaches (*LinkOnly*, *AttrOnly*, and *LinkAsFilter*) on a third real-world dataset in this manner, and show the *LinkAsFilter* clusters achieve a significant improvement in classification accuracy.

5.1 Datasets

We clustered three real-world datasets where attributes exhibit correlation among linked objects, and the link structure exhibits clustering. These are the characteristics we

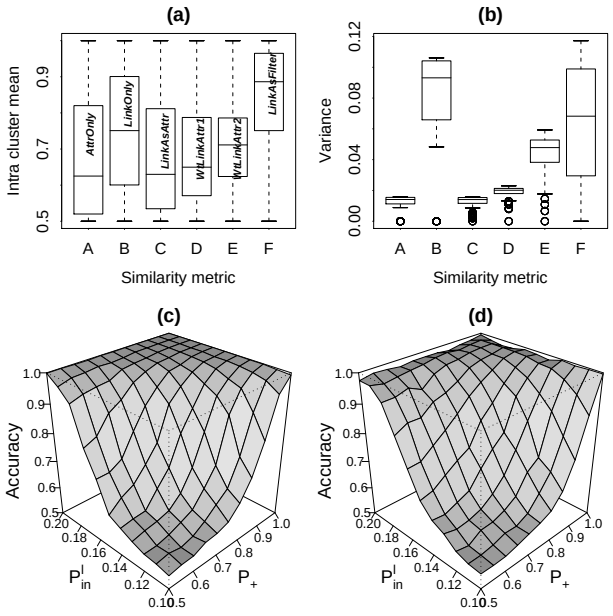

Figure 5: (a) Intra-cluster mean by metric, (b) intra-cluster variance by metric, (c) accuracy of *WtLinkAttr2* and (d) *LinkAsFilter* for 50 objects.

expect to find in datasets that contain communities, and it is in these situations that we expect our clustering algorithm will perform well.

The first data set is drawn from Cora, a database of computer science research papers extracted automatically from the web using machine learning techniques [12]. We selected the largest connected component from the set of machine-learning papers published after 1993. The resulting graph contains 1,042 papers and 2546 citation links. We clustered the undirected version of this graph. The similarity metric considered two topic attributes at different levels of granularity (e.g., {Machine Learning, Neural Networks} and {Planning, Rule Learning}).

The second data set consists of a set of web pages from four computer science departments, collected by the WebKB Project [4]. The web pages have been manually classified into the categories: course, faculty, staff, student, research project, or other. The category "other" denotes a page that is not a home page (e.g., a curriculum vitae linked from a faculty page or homework description linked from a course page). The collection contains approximately 4,000 web pages and 8,000 hyperlinks among those pages. We clustered the largest connected component in these data—a graph of 1236 pages and 3673 hyperlinks. Again, we used the undirected version of the graph. The similarity metric considered two attributes: page category and department. However, the entire component is from a single department (Wisconsin) so the department attribute adds no additional information.

The third data set is a relational data set containing information about the yeast genome at the gene and the protein level (www.cs.wisc.edu/∼dpage/kddcup2001/). The data set contains information about 1,243 genes and 1,734 interactions. We clustered the largest connected component, which consisted of 814 genes and 1475 interactions. The similarity metric considered 13 boolean function attributes. Each gene may have multiple functions. We evaluated the resulting cluster labels' ability to predict gene localization. We applied a relational Bayesian classifier [15] to the entire dataset, using the cluster labels as an additional attribute, and measured performance.

Table 1: Cora cluster examples

Cluster 9: Belief revision: A critique; Plausibility measures and default reasoning; Modeling belief in dynamic systems. Part I: foundations; Knowledge-Based Framework for Belief Change, Part II: Revision and Update; Iterated revision and minimal revision of conditional beliefs; An event-based abductive model of update; On the logic of iterated belief revision; A unified model of qualitative belief change: A dynamical systems perspective; Generalized update: Belief change in dynamic settings
Cluster 14: In defense of C4.5: Notes on learning one-level decision trees; Exploring the decision forest: An empirical investigation of Occams razor in decision tree induction; Algorithmic stability and sanity-check bounds for leave-one-out cross-validation; Bias and the quantification of stability; Characterizing the generalization performance of model selection strategies; A new metric-based approach to model selection; Preventing overfitting of Cross-Validation data; Further experimental evidence against the utility of occams razor
Cluster 19: An empirical evaluation of bagging and boosting; On-line portfolio selection using multiplicative updates; Heterogeneous uncertainty sampling for supervised learning; Improved boosting algorithms using confidence-rated predictions; On-line algorithms in machine learning; Training algorithms for hidden Markov models using entropy based distance functions; A system for multiclass multi-label text categorization; Coevolutionary Search Among Adversaries
Cluster 24: Refinement of Bayesian networks by combining connectionist and symbolic techniques; DistAl: An inter-pattern distance-based constructive learning algorithm; An Anytime Approach to Connectionist Theory Refinement: Refining the Topologies of Knowledge-Based Neural Networks; Creating advice-taking reinforcement learners; Learning controllers for industrial robots; Generating accurate and diverse members of a neural-network ensemble; A Neural Architecture for a High-Speed Database Query System; Comparing methods for refining certainty-factor rule-bases;

5.2 Results

Clustering the sample of Cora papers produced 71 clusters varying in size from 1-202 papers, with an average size of 15. We report statistics for the 28 clusters with more than six papers. Table 1 includes randomly selected titles from four clusters for subjective evaluation. Although we did not use title words in the similarity metrics, the clusters show a surprising uniformity among the titles. This indicates that research papers can be clustered into meaningful groups using the citation structure and topic attributes alone.

To evaluate intra-cluster attribute similarity, we averaged the attribute similarity across all pairs of genes within each cluster. As a baseline measure we calculated the average attribute similarity in ten random clusterings. Figure 6a plots the intra-cluster attribute similarity (dark bars) compared to the expected averages given random clusterings (light bars), with the clusters listed in ascending order by size.

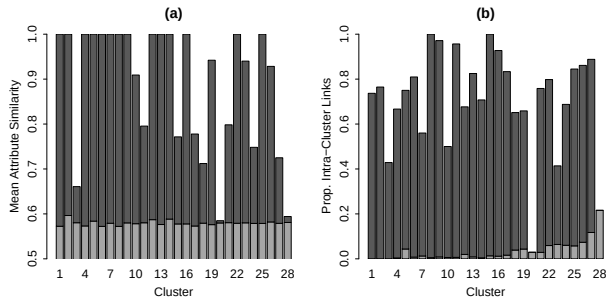

Figure 6: Evaluation of hybrid clusters in Cora.

Table 2: WebKB cluster examples

Cluster 5: http://www.cs.wisc.edu/Dienst/UI/2.0/Describe/-ncstrl.uwmadison/CS-TR-89-890; http://www.cs.wisc.edu/-Dienst/UI/2.0/Describe/ncstrl.uwmadison/CS-TR-90-947; http://www.cs.wisc.edu/Dienst/UI/2.0/Describe/-ncstrl.uwmadison/CS-TR-95-1283; http://www.cs.wisc.edu/-Dienst/UI/2.0/Describe/ncstrl.uwmadison/CS-TR-91-1037; http://www.cs.wisc.edu/Dienst/UI/2.0/Describe/ncstrl.-uwmadison/CS-TR-90-962; http://www.cs.wisc.edu/Dienst/-UI/2.0/Describe/ncstrl.uwmadison/CS-TR-89-900; http://-www.cs.wisc.edu/∼reps/reps.html; http://www.cs.wisc.edu/-Dienst/UI/2.0/Describe/ncstrl.uwmadison/CS-TR-91-1038
Cluster 9: http://www.cs.wisc.edu/∼bart/537/quizzes/-quiz6.html; http://www.cs.wisc.edu/∼bart/cs537.html; http://www.cs.wisc.edu/∼bart/537/quizzes/quiz3.html; http://www.cs.wisc.edu/∼bart/537/quizzes/quiz10.html; http://www.cs.wisc.edu/∼bart/537/quizzes/quiz2.html; http://www.cs.wisc.edu/∼bart/537/programs/program2.html; http://www.cs.wisc.edu/∼bart/537/lecturenotes/-titlepage.html; http://www.cs.wisc.edu/∼bart/537/quizzes/-quiz9.html;
Cluster 11: http://www.cs.wisc.edu/∼cs354-2/cs354/-lec.notes/numbers.html; http://www.cs.wisc.edu/-∼cs354-2/cs354/lec.notes/data.structures.html; http://-www.cs.wisc.edu/∼cs354-2/cs354/solutions/Q2.j.html; http://-www.cs.wisc.edu/∼cs354-2/cs354/lec.notes/arch.features.html; http://www.cs.wisc.edu/∼cs354-2/cs354/lec.notes/-interrupts.html; http://www.cs.wisc.edu/∼cs354-2/cs354/-lec.notes/case.studies.html; http://www.cs.wisc.edu/∼cs354-2/cs354/lec.notes/arith.int.html; http://www.cs.wisc.edu/-∼cs354-2/cs354/lec.notes/MAL.html;
Cluster 14: http://www.cs.wisc.edu/condor/research.html; http://www.cs.wisc.edu/∼bart/cs638.html; http://-www.cs.wisc.edu/coral/coral.people.html; http://-www.cs.wisc.edu/∼brad/brad.html; http://www.cs.wisc.edu/-∼sastry/spring96.html; http://www.cs.wisc.edu/∼ashraf/-ashraf.html; http://maf.wisc.edu/distributed/condor/-index.html; http://www.cs.wisc.edu/∼ssl/resume.html;

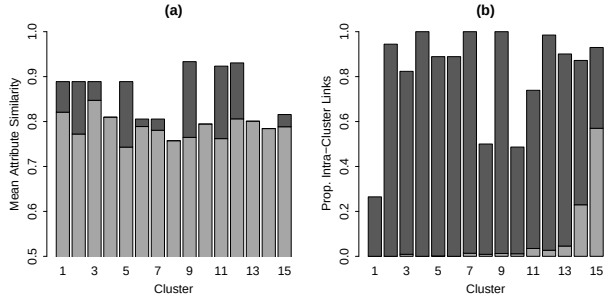

Figure 7: Evaluation of hybrid clusters in WebKB.

Attribute similarity is significantly higher than expected.[2] Note that the largest cluster (#28) does not exhibit high linkage or attribute similarity. This cluster may contain the set of papers that could not be partitioned into smaller clusters (i.e., the papers with no coherent community structure).

Figure 6b shows the actual and expected proportion of intra-cluster citations. To assess the connectivity of the clusters, we compared the proportion of intra-cluster linkage (per cluster) to expected proportions, given ten random clusterings. Again, the proportion of intra-cluster citations is significantly higher than the expected values. This indicates that the clustering technique is finding groups of highly inter-connected research papers.

Clustering the sample of WebKB pages produced 55 clusters varying in size from 1-649 pages, with an average size of 22. We report statistics for the 15 clusters with more than six pages, listed in ascending order by size. Table 2 includes randomly selected URLs from four clusters for subjective evaluation. Recall that the component graph only contains pages from the University of Wisconsin. The selected clusters appear to group by function—for example, tech reports, course pages, or research group pages.

Figure 7b plots the intra-cluster averages compared to the expected averages given random clusterings. Figure 7b shows the actual and expected proportion of intra-cluster hyperlinks. The proportion of intra-cluster linkage is significantly higher than expected, but notice that the largest cluster's (#15) expected linkage is quite high by random chance. This may indicate that the largest cluster contains a set of pages that are too tightly connected to partition. This clustering does exhibit significantly higher than expected attribute similarity. However, we note that the algorithm is still able to cluster pages into groups that are highly interconnected. This indicates that the *LinkAsFilter* metric may be robust to irrelevant attribute values.

Clustering the sample of genes produced 88 clusters varying in size from 1-140 genes, with an average size of 8. We report statistics for the 14 clusters with more than six genes. Intra-cluster attribute similarity (figure 8a) and intra-cluster linkage (figure 8b) are both significantly higher than expected. These results show that the *LinkAsFilter* metric can be used to find groups of genes with similar functions and many common interactions.

The structure of genomic data offers an opportunity for an objective evaluation of the clustering results. Clusters of inter-connected genes with similar associated functions may indicate a group of genes that are interacting to perform a particular function in the cell. If this is the case, the cluster labels should be helpful in predicting gene localization in the cell. To test this hypothesis, we used the cluster labels to predict gene localization. We applied a relational Bayesian classifier (RBC) [15] to the gene data, using the cluster labels as an additional attribute, and measured change in accuracy. Figure 8d reports average 10-fold cross-validation accuracies for RBC models learned using the cluster labels from the *LinkOnly*, *AttrOnly*, and *LinkAsFilter* metrics. The baseline

[2]We assessed significance using two-tailed t-tests, $p < 0.05$.

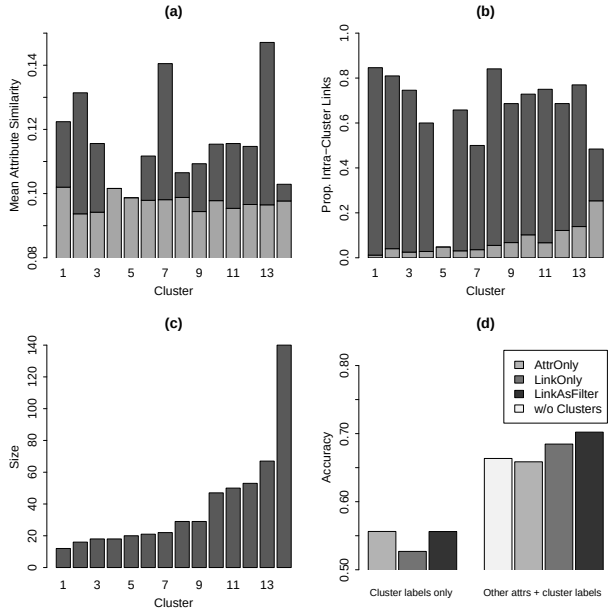

Figure 8: Evaluation of hybrid clusters in Gene.

RBC model used twelve attributes for prediction, including gene phenotype and motif, and achieved an average accuracy of 66.3%. The RBC model that included cluster labels from *AttrOnly* did not significantly improve accuracy.[3] The model that included cluster labels from *LinkOnly* achieved a significant improvement in accuracy, with an average of 68.4%, indicating that gene interactions alone are helpful for predicting location. However, the model that included cluster labels from *LinkAsFilter* achieved an average accuracy of 70.2%. This is a significant improvement over both *LinkOnly* and the baseline RBC model without cluster labels, which demonstrates the utility of clustering for *communities* using both attribute and link information.

6. DISCUSSION

This paper presents a hybrid metric for spectral clustering algorithms that exploits both attribute information and link structure to improve discovery of communities in relational data. There has been relatively little work investigating clustering techniques for relational domains. The work in this area has focused on either complex generative models with latent variables [11, 20, 3], or augmented clustering techniques that use ad-hoc similarity metrics to incorporate both link and attribute information [14, 9]. Due to the complexity of probabilistic relational models with latent variables, and the sparsity of relational graphs that enable the use of efficient eigensolver techniques, we chose to explore extensions to spectral clustering for relational domains.

The most closely related prior work is that of He, Ding, Zha, and Simon [9], which uses a spectral graph-partitioning algorithm to automatically identify topics in sets of retrieved web pages. This approach uses a similarity measure specifically designed for high-dimensional text domains with weighted co-citation links. We differ from this work, and other research on hybrid spectral algorithms, in our exploration of the characteristics that underlie successful similarity metrics.

We have set up a framework to evaluate different similarity metrics quantitatively over a wide range of relational data sets. Our experiments show that increasing the separation between total intra-cluster and inter-cluster transition probabilities results in superior performance over a wide range of data characteristics. One way to increase the separation between cluster transition probabilities is to drop potentially noisy information from consideration. Using this approach, we expect the *LinkAsFilter* metric will successfully recover groupings over a wide range of data characteristics.

There are two primary advantages to using the *LinkAsFilter* metric. The first advantage is algorithm efficiency—there are $O(E)$ approximate eigensolver algorithms, and there are $O(n^{1.4})$ exact eigensolver algorithms for sparse matrices that can exploit the sparse matrix structure produced by the metric. The second advantage is the choice of $\alpha = l$, which is independent of data characteristics. We expect the metric will work well in any dataset exhibiting community structure, provided there is enough data to withstand the associated increase in variance. In small datasets, where the size of the data cannot offset the increase in variance, the application of *balanced* metrics (e.g., *WtLinkAttr2*) may produce superior clusterings. In practice however, this approach is limited by the need to set α to balance the link and attribute information.

With a way to evaluate each setting, an algorithm could search for the best α. Our analysis indicates that the "best" settings will maximize the separation between the intra-cluster and inter-cluster transition probabilities. We conjecture that the eigenvector information—more specifically, the separation between the means of distributions of the eigenvector values on either side of the cut—can be used to approximate this information. We report preliminary findings in support of this conjecture.

Figure 9a graphs the correlation between algorithm performance and the separation of eigenvector-value distributions. We clustered over the space of synthetic datasets described in section 4.1 using 20 different values of α, chosen uniformly in the range $[0, 1]$. We recorded (1) the accuracy of the clustering, and (2) the distance between the means of the eigenvector-value distributions on either side of the chosen cut (after the values were normalized to unit range). Figure 9b shows performance when we set α by maximizing the separation between the means of the eigenvector-value distributions. Comparing this graph to figure 1, we can see that this technique approaches the performance of the *LinkAsFilter* metric. This is a promising direction to explore for applications with little data, where the variance will be too high to apply *LinkAsFilter* successfully.

7. CONCLUSIONS AND FUTURE WORK

We have analyzed the spectral decomposition algorithm from a statistical perspective and shown that the successful hybrid metrics use the link and attribute information to increase the separation between noisy clusters. We have shown an empirical connection between the distribution of tran-

[3]Again, significance was assessed using two-tailed t-tests, $p < 0.05$.

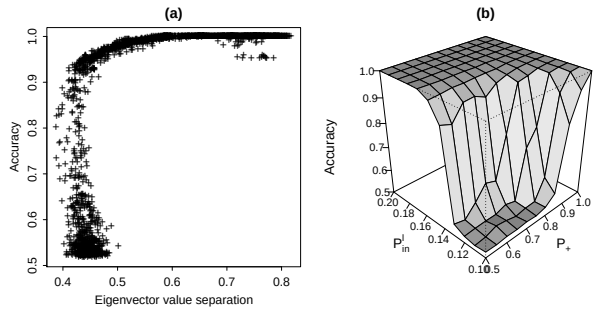

Figure 9: Searching for α to use in the metric: (a) correlation between separation of eigenvector values and accuracy ($corr = 0.71$), and (b) cluster accuracy using α that maximizes separation.

sition probabilities and algorithm performance, connecting both mean and variance to cluster accuracy. Future work will compare this approach to latent-variable relational models and explore complexity/efficiency tradeoffs between the two techniques. Furthermore, we will attempt to derive theoretical bounds on finite-sample performance, and explore the alternative optimization criteria for data with low mean separation, where the NCut criteria prefers high-precision/low-recall groupings.

In addition, the WebKB results suggest an alternative clustering task—clustering data that exhibit *role equivalence* structure, rather than *community* structure. Objects that play the same *roles* in a graph have similar attributes and similar link patterns but may not actually link to each other. For example, faculty pages rarely link to each other but they conistently link to student and course pages. Current methods for grouping data in this manner focus primarily on link information (e.g., [17]). Extending this work to incorporate attribute information seems an exciting direction to explore.

8. ACKNOWLEDGMENTS

The authors acknowledge helpful comments and discussion from Alicia Wolfe. This research is supported under a AT&T Graduate Research Fellowship and by DARPA and AFRL under contract numbers F30602-00-2-0597 and F30602-01-2-0566.

9. REFERENCES

[1] F. Bach and M. Jordan. Learning spectral clustering. In *Proceedings of NIPS16*, 2003.

[2] F. Chung. *Spectral Graph Theory*. The American Mathematical Society, 1997.

[3] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. *Advances in Neural Information Processing Systems*, 10, 2001.

[4] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th National Conference on Artificial Intelligence*, 1998.

[5] I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. of the 7th ACM International Conf. on Knowledge Discovery and Data Mining*, 2001.

[6] W. Donath and A. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17):420–425, 1973.

[7] M. Fiedler. Algebraic connectivity of graphs. *Czecheslovak Math. Jour.*, 23(98):298–305, 1973.

[8] G. Golub and C. V. Loan. *Matrix Computations*. Johns Hopkins University Press, 1983.

[9] X. He, C. Ding, H. Zha, and H. Simon. Automatic topic identification using webpages clustering. In *Proceedings of the 1st IEEE International Conference on Data Mining*, 2001.

[10] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. In *Proceedings of the 41st Symposium on the Foundations of Computer Science*, 2000.

[11] J. Kubica, A. Moore, J. Schneider, and Y. Yang. Stochastic link and group detection. In *Proceedings of the 18th National Conference on Artificial Intelligence*, 2002.

[12] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. A machine learning approach to building domain-specific search engines. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[13] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics*, 2001.

[14] D. Modha and W. Spangler. Clustering hypertext with applications to web searching. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, 2000.

[15] J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational bayesian classifiers. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003.

[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 2001*, 2001.

[17] K. Nowicki and T. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.

[18] B. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Inc., 1980.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[20] B. Taskar, E. Segal, and D. Koller. Probabilistic clustering in relational data. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 2001.

APPENDIX

A. PROOF OF THEOREM

Theorem: *Let $\Delta = (A_1, A_2)$ be a partition of V. Let the function $S(i,j)$ define the similarity measure between $v_i, v_j \in V$. If, $\forall i,j,k$, $S(i,j)$ is conditionally independent of $S(i,k)$ given node i, and $E[\mathbf{P}_{11}]E[\mathbf{P}_{22}] \neq E[\mathbf{P}_{12}]E[\mathbf{P}_{21}]$ then, $\mathbf{P}$ has an eigenvector that will converge to piecewise constant w.r.t. Δ as $|A_1|, |A_2| \to \infty$.*

PROOF. In order to simplify the calculations below, we assume that the two clusters share the same distribution of intra- and inter- cluster similarity values. The symmetry in attribute parameters simplifies the analysis but is not necessary for correctness. Let μ_{in} be the mean intra-cluster similarity for nodes $i,j \in A_1$ or $i,j \in A_2$. Similarly, let μ_{out} be the mean inter-cluster similarity for nodes $i \in A_1$ and $j \in A_2$.

We can represent each entry in $\mathbf{W}$ as a random variable. Consider the entries of row i. The entries $\mathbf{W}_{ij}, \mathbf{W}_{ik}$ are not independent because the similarity values are both based on node i. However, conditioned on the state of i (e.g. attribute values of i), the entries can be viewed as independent random variables if the state of j is independent of the state of k. This assumption corresponds to a generative model in which the objects and links in the graph are conditionally independent given the object cluster memberships.

We will calculate the expected intra- and inter-cluster transition probabilities in $\mathbf{P}$ as a ratio of sums of random variables. Let T^i_{in} be the total intra-cluster transition probability for node i, where $i \in A_{k,k\in 1,2}$, and let $|A_k| = n_k$. Similarly, let T^i_{out} be the total inter-cluster transition probability, and T^i_{all} be the total transition probability. Then $\mathbf{P}^i_{in}$ is the ratio of T^i_{in} and T^i_{all}, and $\mathbf{P}^i_{out}$ is the ratio of T^i_{out} and T^i_{all}.

The normalized transition probabilities in $\mathbf{P}$ then correspond to the ratio of two random variables (e.g., T^i_{in}/T^i_{all}), which can be approximated using a truncated Taylor series expansion. The expectation and variance for intra- and inter-cluster normalized transition probabilities are below. (Analytical derivations are included in Section A.1.)

$$E[\mathbf{P}^i_{in}] = E[T^i_{in}/T^i_{all}] \approx \frac{\mu_{T_{in}}}{\mu_{T_{all}}} \cdot [1 + [\frac{\sigma_{T_{all}}}{\mu_{T_{all}}}]^2 - \frac{\sigma_{T_{in}T_{all}}}{\mu_{T_{in}}\mu_{T_{all}}}]$$

$$E[\mathbf{P}^i_{out}] = E[T^i_{out}/T^i_{all}] \approx \frac{\mu_{T_{out}}}{\mu_{T_{all}}} \cdot [1 + [\frac{\sigma_{T_{all}}}{\mu_{T_{all}}}]^2 - \frac{\sigma_{T_{out}T_{all}}}{\mu_{T_{out}}\mu_{T_{all}}}]$$

where σ_{XY} is the covariance of X, Y.

As $n_1, n_2 \to \infty$, it follows directly from the *Law of Large Numbers* that the value of $T^i_{in}/T^j_{in} \to 1$ for $i,j \in A_k$, since T_{in} is a sum of independent random variables with finite mean and variance. A similar argument holds for T_{out} and T_{all}. Now consider the normalized transition probabilities for $\mathbf{P}$. If, in the limit, the sums T^i_{in} (and T^i_{out}, T^i_{all}) converge to the same value for all $i \in A_k$, then the normalized sums $\mathbf{P}^i_{in}$ will converge to the same value $\mathbf{P}_{in}$ for all $i \in A_k$. A similar argument holds for $\mathbf{P}^i_{out}$.

As $n_1, n_2 \to \infty$, we can decompose the matrix $\mathbf{P}$ into $\mathbf{P} = \mathbf{P'} + \epsilon\mathbf{E}$, where $\mathbf{P'}$ is a matrix with constant transition probabilities $\mathbf{P}_{in}$ and $\mathbf{P}_{out}$, and $\mathbf{E}$ is a perturbation matrix with $||\mathbf{E}||_2 = 1$. Then by matrix perturbation theory [8]:

$$(\mathbf{P'} + \epsilon\mathbf{E})\mathbf{x}_i(\epsilon) = \lambda_i(\epsilon)\mathbf{x}_i(\epsilon)$$

$$where \;\; \mathbf{x}_i(\epsilon) = \mathbf{x}_i + \epsilon \textstyle\sum_{j=1, j\neq i}^{n} \{\frac{\mathbf{y}_j{}^T \mathbf{E}\mathbf{x}_i}{(\lambda_i - \lambda_j)\mathbf{y}_j{}^T\mathbf{x}_i}\} + O(\epsilon^2)\,,$$

$$and \quad \lambda_i(\epsilon) = \lambda_i \pm \frac{\epsilon}{|\mathbf{y}_i{}^T\mathbf{x}_i|}$$

Here $\mathbf{x}_i$, $\mathbf{y}_i$, and λ_i, are the right and left eigenvectors, and the eigenvalues of $\mathbf{P'}$. As $n_1, n_2 \to \infty$, $\epsilon \to 0$ and the eigenvectors of $\mathbf{P}$ will converge to the eigenvectors of $\mathbf{P'}$. Therefore the graph will converge to a Markov chain with state space $\Delta = (A_1, A_2)$, and constant transition probabilities $\mathbf{R}_{11} = \mathbf{R}_{22} = E[\mathbf{P}^i_{in}]$, and $\mathbf{R}_{12} = \mathbf{R}_{21} = E[\mathbf{P}^i_{out}]$. If $\mathbf{R}_{11} \neq \mathbf{R}_{12}$, then $\mathbf{R}$ will be non-singular, and by proposition 2 in [13], $\mathbf{P}$ will have a piecewise linear eigenvector w.r.t Δ. □

A.1 Analytic Derivations

When $S(i,j)$ is conditionally independent of $S(i,k)$ given the state of node i, the cluster transition probabilities are simply sums of independent random variables. Using conditional expectation ($E[h(X,Y)] = E_X\{E[h(X,Y)|X]\}$), we can calculate the expectation for T^i_{in} based on the state of i, which we refer to as i_S:

$$\begin{aligned}
E[T^i_{in}] &= E[\textstyle\sum_{j\in A_k} S(i,j)] \\
&= \textstyle\sum_{i_S} p(i_S) \cdot E[\sum_{j\in A_k} S(i_S, j)] \\
&= \textstyle\sum_{i_S} p(i_S) \cdot n_k \cdot E[S(i_S, j) | j \in A_k] \\
&= n_k \cdot \textstyle\sum_{i_S} p(i_S) \cdot \sum_{j_S} p(j_S) \cdot S(i_S, j_S) \\
&= n_k \cdot \textstyle\sum_{i_S} \sum_{j_S} p(i_S) \cdot p(j_S) \cdot S(i_S, j_S) \\
&= n_k \cdot E[S_{in}] \\
&= n_k \cdot \mu_{in}
\end{aligned}$$

Total inter-cluster and overall means are calculated in a similar fashion. $E[T^i_{out}] = n_{k'} \cdot \mu_{out}$, and $E[T^i_{all}] = (n_k \cdot \mu_{in}) + (n_{k'} \cdot \mu_{out})$, where $n_{k'} = n_{i,i\neq k}$.

The variance of the total intra-cluster similarity is calculated as follows [4]:

$$\begin{aligned}
Var[T^i_{in}] &= Var[\textstyle\sum_{j\in A_k} S(i,j)] \\
&= E_{i_S}\{Var[\textstyle\sum_{j\in A_k} S(i_S, j)]\} \\
&= \textstyle\sum_{i_S} p(i_S) \cdot Var[\sum_{j\in A_k} S(i_S, j)] \\
&= \textstyle\sum_{i_S} p(i_S) \cdot n_k \cdot Var[S(i_S, j) | j \in A_k] \\
&= n_k \cdot \textstyle\sum_{i_S} \sum_{j_S} p(i_S) \cdot p(j_S) \cdot \{S(i_S, j_S) - E_{i_S}[S(i_S, j_S)]\}^2
\end{aligned}$$

Total inter-cluster and overall variance are calculated in a

[4]The derivation uses the following equivalence:

$$\begin{aligned}
Var(h(X,Y)) &= E[h(X,Y)^2] - E[h(X,Y)]^2 \\
&= E_X\{E[h(X,Y)^2|X]\} - E_X\{E[h(X,Y)|X]^2\} \\
&= E_X\{Var(h(X,Y)|X)\}
\end{aligned}$$

similar fashion: $Var[T^i_{out}] = n_{k'} \cdot \sum_{i_S} p(i_S) \cdot Var[S(i_S, j)|j \in A_{k'}]$, and $Var[T^i_{all}] = \sum_{i_S} p(i_S) \{n_{k'} \cdot Var[S(i_S, j)|j \in A_{k'}]$ $+n_k \cdot Var[S(i_S, j)|j \in A_k]\}$.

From these we can calculate the expected transition probabilities of $\mathbf{P}$ using the ratio of two random variables (e.g., T_{in}/T_{all}). These calculations use an approximation of the ratio of two random variables, based on a truncated Taylor series expansion:

$$\begin{aligned} E[X/Y] &\approx \tfrac{\mu_X}{\mu_Y} \cdot [1 + [\tfrac{\sigma_Y}{\mu_Y}]^2 - \tfrac{\sigma XY}{\mu_X \mu_Y}] \\ Var(X/Y) &\approx [\tfrac{\mu_X}{\mu_Y}]^2 \cdot [[\tfrac{\sigma_X}{\mu_X}]^2 + [\tfrac{\sigma_Y}{\mu_Y}]^2 - 2\tfrac{\sigma XY}{\mu_X \mu_Y}] \end{aligned}$$

The expectation and variance for intra- and inter-cluster normalized transition probabilities are as follows:

$$\begin{aligned} E[\mathbf{P}^i_{in}] &= E[T^i_{in}/T^i_{all}] &&\approx \tfrac{\mu_{T_{in}}}{\mu_{T_{all}}} \cdot [1 + [\tfrac{\sigma_{T_{all}}}{\mu_{T_{all}}}]^2 - \tfrac{\sigma_{T_{in}T_{all}}}{\mu_{T_{in}}\mu_{T_{all}}}] \\ Var[\mathbf{P}^i_{in}] &= Var[T^i_{in}/T^i_{all}] &&\approx [\tfrac{\mu_{T_{in}}}{\mu_{T_{all}}}]^2 \cdot [[\tfrac{\sigma_{T_{in}}}{\mu_{T_{in}}}]^2 + [\tfrac{\sigma_{T_{all}}}{\mu_{T_{all}}}]^2 - 2\tfrac{\sigma_{T_{in}T_{all}}}{\mu_{T_{in}}\mu_{T_{all}}}] \\ E[\mathbf{P}^i_{out}] &= E[T^i_{out}/T^i_{all}] &&\approx \tfrac{\mu_{T_{out}}}{\mu_{T_{all}}} \cdot [1 + [\tfrac{\sigma_{T_{all}}}{\mu_{T_{all}}}]^2 - \tfrac{\sigma_{T_{out}T_{all}}}{\mu_{T_{out}}\mu_{T_{all}}}] \\ Var[\mathbf{P}^i_{out}] &= Var[T^i_{out}/T^i_{all}] &&\approx [\tfrac{\mu_{T_{out}}}{\mu_{T_{all}}}]^2 \cdot [[\tfrac{\sigma_{T_{out}}}{\mu_{T_{out}}}]^2 + [\tfrac{\sigma_{T_{all}}}{\mu_{T_{all}}}]^2 - 2\tfrac{\sigma_{T_{out}T_{all}}}{\mu_{T_{out}}\mu_{T_{all}}}] \end{aligned}$$

where σ_{XY} is the covariance of X, Y. For the equations above, the covariance of T_{in} and T_{all} reduces to the variance of T_{in}, using conditional expectation to eliminate the covariance:

$$\begin{aligned} \sigma_{T_{in}T_{all}} &= E[T_{in}T_{all}] - E[T_{in}] \cdot E[T_{all}] \\ &= E[T_{in}(T_{in} + T_{out})] - E[T_{in}] \cdot E[(T_{in} + T_{out})] \\ &= E[T^2_{in} + T_{in} \cdot T_{out}] - E[T_{in}]^2 - E[T_{in}] \cdot E[T_{out}] \\ &= E[T^2_{in}] + E[T_{in} \cdot T_{out}] - E[T_{in}]^2 - E[T_{in}] \cdot E[T_{out}] \\ &= E[T^2_{in}] - E[T_{in}]^2 + E[T_{in} \cdot T_{out}] - E[T_{in}] \cdot E[T_{out}] \\ &= Var(T_{in}) - \textstyle\sum_{i_S} p(i_S)\{E[T_{in} \cdot T_{out}|i] - E[T_{in}|i] \cdot E[T_{out}|i]\} \\ &= Var(T_{in}) - \textstyle\sum_{i_S} p(i_S) \cdot 0 \\ &= Var(T_{in}) \end{aligned}$$

A similar derivation applies to the covariance of T_{out} and T_{all}.